Metaheuristics for the Bi-objective Ring Star Problem

Arnaud Liefooghe, Laetitia Jourdan, Matthieu Basseur, El-Ghazali Talbi, Edmund K. Burke

Metaheuristics for the Bi-objective Ring Star Problem

Arnaud Liefooghe[1], Laetitia Jourdan[1], Matthieu Basseur[2], El-Ghazali Talbi[1], and Edmund K. Burke[2]

[1] Laboratoire d'Informatique Fondamentale de Lille (LIFL), INRIA, CNRS
Université des Sciences et Technologies de Lille
Parc scientifique de la Haute Borne, 40 avenue Halley, Bât. A, Park Plaza
59650 Villeneuve d'Ascq, France
{Arnaud.Liefooghe,Laetitia.Jourdan,El-Ghazali.Talbi}@lifl.fr
[2] School of Computer Science,
University of Nottingham, Jubilee Campus, Wollaton Road,
Nottingham, NG8 1BB, United Kingdom
Matthieu.Basseur@info.univ-angers.fr,ekb@cs.nott.ac.uk

Abstract. The bi-objective ring star problem aims to locate a cycle through a subset of nodes of a graph while optimizing two types of cost. The first criterion is to minimize a ring cost, related to the length of the cycle, whereas the second one is to minimize an assignment cost, from non-visited nodes to visited ones. In spite of its natural multi-objective formulation, this problem has never been investigated in such a way. In this paper, three metaheuristics are designed to approximate the whole set of efficient solutions for the problem under consideration. Computational experiments are performed on well-known benchmark test instances, and the proposed methods are rigorously compared to each other using different performance metrics.

1 Introduction

The purpose of the bi-objective ring star problem is to find a cycle (the ring) which visits a subset of nodes of a graph. The two objectives are the minimization of a cost associated to the ring itself and the minimization of a cost associated to the arcs directed from non-visited nodes to visited ones. Although this problem is clearly bi-objective, it has always been investigated in a single-objective way. It was introduced by Labbé et al. [12], where the goal was to minimize the sum of both costs. Another mono-criterion formulation of the problem, where one of the objectives is regarded as a constraint, has been investigated, for instance, by Renaud et al. [16]. These two formulations are commonly used to convert a multi-objective problem into a single-objective one by using scalar approaches [14].

The ring star problem has a wide range of industrial applications, including telecommunication networks design, school bus routing, routing of essential medical care services, circular-shaped transportation, and post-box location. However, in spite of its real-world applications, this is the first time that such a problem is studied in a bi-objective way, perhaps because of its complexity. Indeed, it is particularly challenging because, once it is decided which nodes have to be visited or not, a classical travelling salesman problem still remains to be solved. Nevertheless, Current and Schilling [5] investigated two variants of a similar problem: the *median tour problem* and the *maximal covering tour problem*. In both, one criterion is the minimization of the total length of the tour, while another one is the maximization of the access to the tour for non-visited nodes. To tackle these problems, the authors used a kind of lexicographic method, where one objective function is optimized after another. Furthermore, Dorner et al. [8] recently formulated a problem of tour planning for mobile healthcare facilities in Senegal. A mobile facility has to visit a subset of nodes. Non-visited nodes are then assigned to their closest tour stop or are regarded as unable to reach a tour stop. The obectives are the minimization of the ratio between medical working time and total working time, the minimization of the average distance to the nearest tour stops, and the maximization of a coverage criterion. The authors designed a Pareto ant colony optimization algorithm and two multi-objective genetic algorithms to solve real-world instances.

In this paper, we investigate metaheuristic solution methods for the problem under consideration. Three metaheuristics are designed to approximate the whole set of efficient solutions. A population-based local search and two evolutionary algorithms are compared on state-of-the-art instances involving up to 300 nodes. The reminder of the paper is organized as follows. In Section 2, we provide the basic definitions for multi-objective optimization and the formulation of the bi-objective ring star problem. Section 3 presents three resolution methods designed to tackle the problem under consideration. Some experimental results and a comparative study are provided in Section 4, while the last section concludes the paper and discusses perspectives about this work.

2 Preliminaries

This section presents some basic concepts related to multi-objective optimization and provides the formulation of the bi-objective ring star problem.

2.1 Multi-objective Optimization

A *Multi-objective Optimization Problem* (MOP) aims to optimize a set of $n \geq 2$ objective functions $f_1, f_2, \ldots, f_n$ simultaneously. Without loss of generality, we assume that all n objective functions have to be minimized. Let X denote the set of feasible solutions in the *decision space*, and Z the set of feasible points in the *objective space*. To each decision vector $x \in X$ is assigned exactly one objective vector $z \in Z$ on the basis of a vector function $f : X \rightarrow Z$ with $z = f(x) = (f_1(x), f_2(x), \ldots, f_n(x))$. In the case of a *Multi-objective Combinatorial Optimization Problem* (MCOP), note that a decision vector $x \in X$ has a finite set of possible values.

Definition 1. *An objective vector* $z \in Z$ weakly dominates *another objective vector* $z' \in Z$ *if and only if* $\forall i \in [1..n]$, $z_i \leq z'_i$.

Definition 2. *An objective vector* $z \in Z$ dominates *another objective vector* $z' \in Z$ *if and only if* $\forall i \in [1..n]$, $z_i \leq z'_i$ *and* $\exists j \in [1..n]$ *such as* $z_j < z'_j$.

Definition 3. *An objective vector* $z \in Z$ *is* non-dominated *if and only if there does not exist another objective vector* $z' \in Z$ *such that* z' *dominates* z.

A solution $x \in X$ is said to be *efficient* (or *Pareto optimal*) if its mapping in the objective space results in a non-dominated point. The set of all efficient solutions is the *efficient* (or *Pareto optimal*) *set*, denoted by X_E. The set of all non-dominated vectors is the *non-dominated front* (or the *trade-off surface*), denoted by Z_N. A common approach in solving MOPs is to find or to approximate the set of efficient solutions; or at least a solution $x \in X_E$ for each non-dominated vector $z \in Z_N$ such as $f(x) = z$. A reasonable basic introduction to multi-objective optimization can be found in [6].

Note that we will assume, throughout the paper, that objective values are normalized. To achieve this, the minimum and the maximum value of each objective function are used in order to adaptively replace each objective function by its corresponding normalized function, so that its values lie in the interval $[0, 1]$.

2.2 The Bi-objective Ring Star Problem

The *Ring Star Problem* (RSP) can be described as follows. Let $G = (V, E, A)$ be a complete mixed graph where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of vertices, $E = \{[v_i, v_j] | v_i, v_j \in V, i < j\}$ is a set of edges, and $A = \{(v_i, v_j) | v_i, v_j \in V\}$ is a set of arcs. Vertex v_1 is the depot. To each edge $[v_i, v_j]$ we assign a non-negative *ring cost* c_{ij}, and to each arc (v_i, v_j) we assign a non-negative *assignment cost* d_{ij}. The RSP consists of locating a simple cycle through a subset of nodes $V' \subset V$ (with $v_1 \in V'$) while (i) minimizing the sum of the ring costs related to all edges that belong to the cycle, and (ii) minimizing the sum of the assignment costs of arcs directed from every non-visited node to a visited one so that the associated cost is minimum. An example of solution is given in Figure 1, where solid lines represent edges that belong to the ring and dashed lines represent arcs of the assignments. The first objective is called the *ring cost* and is defined as:

$$\sum_{[v_i, v_j] \in E} c_{ij} x_{ij} \ , \tag{1}$$

where x_{ij} is a binary variable equal to 1 if and only if the edge $[v_i, v_j]$ belongs to the cycle. The second objective, the *assignment cost*, can be computed as follows:

$$\sum_{v_i \in V \setminus V'} \min_{v_j \in V'} d_{ij} \ . \tag{2}$$

The RSP is an NP-hard combinatorial problem since the particular case of visiting the whole set of nodes is equivalent to a traditional travelling salesman problem.

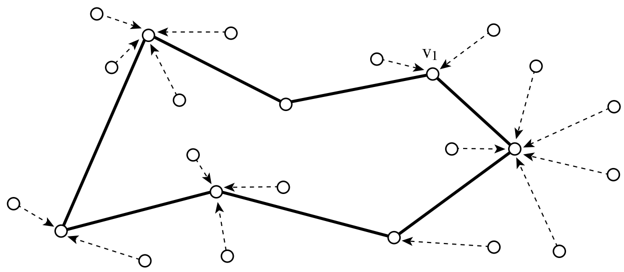

Fig. 1. An example of a solution for the ring star problem.

3 Metaheuristics for the Bi-objective Ring Star Problem

Three metaheuristics are proposed to tackle the bi-objective RSP: a variable neighbourhood iterative Local Search (LS) and two Evolutionary Algorithms (EAs). These algorithms are respectively steady-state variations of IBMOLS [1], IBEA [17] and NSGA-II [7]. IBMOLS and IBEA are both recent indicator-based metaheuristics, whereas NSGA-II is one of the most often used Pareto-based resolution methods. In this section, RSP-specific components are described after we have presented the main characteristics of the LS and of the EAs.

3.1 A Multi-objective Local Search

Since they are easily adaptable to the multi-objective context, many of the search algorithms proposed to tackle MOPs are EAs. However, LS algorithms are known to be effective metaheuristics for solving real-world applications [4, 9]. Several multi-objective LS approaches have been proposed in the literature. In particular, the Indicator-Based Multi-Objective Local Search (IBMOLS for short) has recently been presented in [1]. IBMOLS is a generic population-based multi-objective LS dealing with a fixed population size. This allows to obtain a set of efficient solutions in a single run without specifying any mechanism to control the number of solutions during the search process. Moreover, IBMOLS represents an alternative to aggregation- and Pareto-based multi-objective metaheuristics. Indeed, as proposed in [17], it is assumed that the optimization goal is given in terms of a binary quality indicator I [19] which can be regarded as an extension of the Pareto dominance relation. A value $I(A, B)$ quantifies the difference in quality between two non-dominated sets A and B. So, if Z_N denotes the optimal non-dominated front, the overall optimization goal can be formulated as:

$$\arg \min_{A \in \Omega} I(A, Z_N) \ , \tag{3}$$

where Ω denotes the space of all non-dominated set approximations. As noted in [17], Z_N does not have to be known, it is just required in the formalization of the optimization goal. Since Z_N is fixed, I actually represents a unary function that assigns a real number reflecting the quality of each approximation set.

One of the main advantages of indicator-based optimization is that no specific diversity preservation mechanism is generally required, according to the indicator being used.

The IBMOLS algorithm maintains a population P. Then, it generates the neighbourhood of a solution contained in P until a good solution is found (*i.e.* one that is better than at least one solution of P in terms of the indicator being used). By iterating this simple principle to every solution of P, we obtain a local search step. The whole local search stops when the archive of potentially efficient solutions has not received any new item during a complete local search step. Moreover, as local search methods are usually performed in an iterative way, a population re-initialization scheme has to be designed after each local search. Several strategies can be used within an iterative IBMOLS [1]. Solutions can be re-initialized randomly, and crossover or random noise can be applied to solutions of the efficient set approximation. The interested reader could refer to [1] for more details about IBMOLS.

A benefical feature of this LS is the low number of parameters that are required. In addition to the population size, the binary quality indicator to be used and the population re-initialization strategy (between each local search) are the two only other problem-independent parameters. Indeed, several quality indicators can be used within IBMOLS. The binary additive ϵ-indicator [17] is particularly well-adapted to indicator-based search and seems to be efficient on different kinds of problems (see, for instance, [1, 17]). It is capable of obtaining both a well-converged and a well-diversified Pareto set approximation. This indicator computes the minimum value by which a solution $x_1 \in X$ can be translated in the objective space to weakly dominate another solution $x_2 \in X$. For a minimization problem, it is defined as follows:

$$I_{\epsilon+}(x_1, x_2) = \max_{i \in \{1,\dots,n\}} (f_i(x_1) - f_i(x_2)) \ . \tag{4}$$

Furthermore, to evaluate the quality of a solution according to a whole population P and a binary quality indicator I, different approaches exist. As proposed in [17], we will here consider an additive technique that amplifies the influence of solutions mapping to dominating points over solutions mapping to dominated ones which can be outlined as follows:

$$I(P \setminus \{x\}, x) = \sum_{x^\star \in P \setminus \{x\}} -e^{-I(x^\star, x)/\kappa} \ , \tag{5}$$

where $\kappa > 0$ is a scaling factor. However, the initial experiments were not satisfactory because the algorithm was not able to find the extreme points of the trade-off surface. This is known to be one of the drawbacks of the ϵ-indicator, apparently due to the high convexity of the front. To tackle this problem, we add a condition preventing the deletion of solutions corresponding to the extreme non-dominated vectors during the replacement step of IBMOLS. Additionally, the population re-initialization scheme used between each local search is based on random noise, such as in the basic simulated annealing algorithm [4]. Random noise consists of multiple mutations applied to N different randomly chosen solutions contained in the archive of potentially efficient solutions. If the size of the archive is less than N, the population is filled with random solutions.

3.2 Multi-objective Evolutionary Algorithms

The multi-objective EAs designed for the RSP are variations of two state-of-the-art methods, namely IBEA [17] and NSGA-II [7]. Some minor modifications have been carried out to improve the algorithms for the particular case of the addressed problem, for which the set of non-dominated points is, in general, very large.

IBEA Introduced by Zitzler and Künzli [17], the *Indicator-Based Evolutionary Algorithm* (IBEA) is, like IBMOLS, an indicator-based metaheuristic. The fitness assignment scheme of this EA is based on a pairwise comparison of solutions contained in a population by using a binary quality indicator. As within IBMOLS, no diversity preservation technique is required, according to the indicator being used. The selection scheme for reproduction is a binary tournament between randomly chosen individuals. The replacement strategy is an environmental one that consists of deleting, one-by-one, the worst individuals, and in updating the fitness values of the remaining solutions each time there is a deletion; this is continued until the required population size is reached. Moreover, an archive stores solutions mapping to potentially non-dominated points, in order to prevent their loss during the stochastic search process. However, in our case, and in contrast to the IBEA defined in [17], this archive is updated at each generation since the beginning of the EA, so that the output size is not necessarily less than or equal to the population size. Just like for the IBMOLS algorithm, the indicator used within IBEA is the additive ϵ-indicator; and the same mechanism has been used to prevent the loss of the extreme points on the trade-off surface.

NSGA-II At each generation of NSGA-II (*Non-dominated Sorting Genetic Algorithm II* [7]), the solutions contained in the population are ranked into several classes. Individuals mapping to vectors from the first front all belong to the best efficient set; individuals mapping to vectors from the second front all belong to the second best efficient set; and so on. Two values are then computed for every solution of the population. The first one corresponds to the *rank* the corresponding solution belongs to, and represents the quality of the solution in terms of convergence. The second one, the *crowding distance*, consists of estimating the density of solutions surrounding a particular point of the objective space, and represents the quality of the solution in terms of diversity. A solution is said to be better than another if it has the best rank, or in the case of a tie, if it has the best crowding distance. The selection strategy is a deterministic tournament between two random solutions. At the replacement step, only the best individuals survive, with respect to the population size. Likewise, an external population is added to the steady-state NSGA-II in order to store every potentially efficient solution found during the search.

3.3 Application to the Bi-objective Ring Star Problem

This section provides the problem-specific steps of the metaheuristics introduced earlier. Components designed for the particular case of the bi-objective RSP, such as the encoding mechanism, the population initialization as well as the neighbourhood, mutation and crossover operators, are described below.

Solution Encoding The encoding mechanism used to represent a RSP solution, for both the LS and the EAs, is based on the random keys concept proposed by Bean [2]. This implementation has already been successfully applied for a single-objective version of the RSP in [16]. To each node v_i belonging to the ring we assign exactly one *random key* $x_i \in [0, 1[$, where $x_1 = 0$. A special value is assigned to unvisited nodes. Thus, the ring route associated to a solution corresponds to the nodes ordered according to their random keys in an increasing way; *i.e.* if $x_i < x_j$, then v_j comes after v_i. As an example, a possible representation for the cycle (v_1,v_7,v_4,v_9,v_2,v_6) is given in Figure 2. Vertices v_3,v_5,v_8 and v_{10} are assigned to a visited node in such a way that the associated assignment cost is minimum.

Vertex	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
Random key	0	0.7	-	0.3	-	0.8	0.2	-	0.5	-

Fig. 2. A RSP solution represented by random keys.

Population Initialization For every optimization method, the initial population has been generated randomly. Each node has a probability $p = 0.5$ that it will be visited or not, and to each visited vertex we associate a key randomly generated between 0 and 1.

Neighbourhood and Mutation Operators As the RSP is both a routing problem and an assignment problem, different neighbourhood and mutation operators have to be designed. Here, we consider the following:

- *insert operator*: adds an unvisited node v_i in the cycle, the position where to insert v_i is chosen in order to minimize the ring cost
- *remove operator*: removes a vertex v_j of the ring
- *2-opt operator*: applies a 2-opt operator between two nodes of the cycle v_i and v_j, *i.e.* it reverses the sequence of visited nodes between v_i and v_j.

For the LS, the neighbours of a solution are randomly explored, without considering any order between these three operators; and each neighbour is at most visited once. Moreover, note that it is not necessary to completely re-evaluate a solution each time a neighbourhood operator is applied. Thus, after an *insert* neighbourhood operator, we just have to re-assign unvisited nodes in order to minimize the assignment cost. After a *remove* neighbourhood operator, we just have to re-assign the nodes that were previously assigned to the one that has been removed. And, after a *2-opt* neighbourhood operator, we just have to recompute the ring cost, the assignment cost being unchanged. In the case of mutations, the operators are applied to randomly chosen vertices.

Crossover Operator The crossover operator is a quadratic crossover closely related to the one proposed in [16]. Two randomly selected solutions s_1 and s_2 are divided in a particular position. Then, the first part of s_1 is combined with the second part of s_2 to build a first offspring, and the first part of s_2 is combined with the second part of s_1 to build a second offspring. Every node retains its random key so that it enables an easy reconstruction of the new individuals. Thanks to the random keys encoding mechanism, solutions having a different ring size can easily be recombined, even if the initial ring structures are generally broken in the offspring solutions.

4 Experiments

The metaheuristics described in the previous section have all been implemented using the ParadisEO-MOEO library[3] [13]. ParadisEO-MOEO is a C++ white-box object-oriented framework dedicated to the reusable design of metaheuristics for multi-objective optimization. All the algorithms share the same base components for a fair comparison between them. Computational runs were performed on an Intel Core 2 Duo 6600 (2×2.40 GHz) machine, with 2 GB RAM.

4.1 Experimental Protocol

Benchmarks The performance of the metaheuristics has been tested on different instances taken from the TSPLIB[4] [15]. These instances involve between 50 and 300 nodes. The number at the end of an instance's name represents the number of nodes for the instance under consideration. Let l_{ij} denote the distance between two nodes v_i and v_j of a TSPLIB file. Then, the ring cost c_{ij} and the assignment cost d_{ij} have both been set to l_{ij} for every pair of nodes v_i and v_j.

Parameter Setting For each one of the metaheuristics proposed to tackle the bi-objective RSP, the search process stops after a certain ammount of run time. As shown in Table 1, this run time is defined according to the size of the instance under consideration. Likewise, the population size depends on the number of vertices involved in the instance (see Table 1). For each instance, a small (S), a medium (M), a large (L) and an extra-large (XL) population size have been tested. The noise rate for the population re-initialization in the iterated version of IBMOLS is set to a fixed percentage of the instance's size. We investigate

[3] ParadisEO is available at `http://paradiseo.gforge.inria.fr`.

[4] Benchmarks are available at `http://elib.zib.de/pub/mp-testdata/tsp/tsplib`.

three different values for this noise rate: 5%, 10% and 20%. Then, $0.05 \times n$, $0.1 \times n$ and $0.2 \times n$ random mutations are applied respectively for a problem with n nodes. For both IBMOLS and IBEA, the scaling factor κ is set to 0.05. Finally, for the EAs, the crossover probability is set to 0.25, and the mutation probability to 1.00, with a probability of 0.25, 0.25 and 0.50 for the *remove*, the *insert* and the *2-opt* operator, respectively.

Table 1. Instance-dependant parameters setting.

Instance	Population size				Running time
	S	M	L	XL	
eil51	5	10	15	100	20"
st70	5	10	15	100	1'
kroA100	10	15	20	100	2'
bier127	10	15	20	100	5'
kroA150	15	20	30	100	10'
kroA200	15	20	30	100	20'
pr264	15	20	30	100	50'
pr299	15	20	30	100	100'

Performance Assessment For each TSPLIB instance and each metaheuristic proposed in Section 3, a set of 20 runs, with different initial populations, has been performed. In order to evaluate the quality of the non-dominated front approximations obtained for a specific test instance, we follow the protocol given in [11]. First, we compute a reference set $Z_N^\star$ of non-dominated points extracted from the union of all these fronts. Second, we define $z^{max} = (z_1^{max}, z_2^{max})$, where z_1^{max} (respectively z_2^{max}) denotes the upper bound of the first (respectively second) objective in the whole non-dominated front approximations. Then, to measure the quality of an output set A in comparison to $Z_N^\star$, we compute the difference between these two sets by using the unary hypervolume metric [18], $(1.05 \times z_1^{max}, 1.05 \times z_2^{max})$ being the reference point. The hypervolume difference indicator (I_H^-) computes the portion of the objective space that is weakly dominated by $Z_N^\star$ and not by A. Furthermore, we also consider the R2 indicator proposed in [10] with a Chebycheff utility function defined by $z^\star = (1,1)$, $\rho = 0.01$ and a set Λ of 500 uniformly distributed normalized weighted vectors. As a consequence, for each test instance, we obtain 20 hypervolume differences and 20 R2 measures, corresponding to the 20 runs, per algorithm. As suggested by Knowles et al. [11], once all these values are computed, we perform a statistical analysis on pairs of optimization methods for a comparison on a specific test instance. To this end, we use the Mann-Whitney statistical test as described in [11], with a p-value lower than 5%. Note that all the performance assessment procedures have been achieved using the performance assessment tool suite provided in PISA[5] [3].

[5] The package is available at `http://www.tik.ee.ethz.ch/pisa/assessment.html`.

4.2 Computational Results and Discussion

Table 2 presents the results obtained by the metaheuristics on eight different test instances. Due to space limitation and in order to simplify the reading of the table, only the results obtained by a large population size and by a noise rate of 5% for IBMOLS and by an extra-large population size for NSGA-II and IBEA are reported in the paper. These parameters have respectively been chosen as they were globally more efficient for each one of the algorithms. Overall, with respect to the metrics we used, we can see that IBMOLS performs significantly better than IBEA and NSGA-II on most test instances. Nevertheless, it is not the case on large problems (*pr264* and *pr299*), where IBMOLS is outperformed by both algorithms according to the R2 metric. Additionally, although IBEA is in general statistically outperformed by IBMOLS, it performs significantly better than NSGA-II on a large number of the tested instances, and never performs significantly worse on each one of them. Furthermore, we can see that the overall efficiency of NSGA-II is very poor since it is statistically outperformed on most problems, except occasionally where it performs better than the IBMOLS algorithm, as pointed out above. To summarise, IBMOLS performs well on small-size RSP instances, but seems to have more trouble in dealing with large ones. Moreover, we also compared our results to the ones given in [12] for a mono-objective version of the problem. For each test instance, the error ratio between the point belonging to $Z_N^{\star}$ that minimizes the single objective function investigated in [12] and the exact optimal value is is always under 2% and is averagely under 0.5%.

One of the main characteristics of the problem under consideration seems to be the high number of points located in the trade-off surface. Then, after a certain number of iterations, a large part of the population involved in all the algorithms might map to potentially non-dominated points. This could explain the low efficiency of NSGA-II. Since the same rank is assigned to the major part of the population, only the crowding distance is used to compare solutions. However, the indicator-based fitness assignment scheme is obviously much more suited to determine potentially efficient solutions than the single crowding distance. Moreover, the high performance of IBMOLS in comparison to IBEA might depends on how close are the solutions which map to non-dominated points in the decision space. If these solutions are close to each other according to the neighbourhood operators, a LS is known to be particularly well-suited to find additional interesting solutions by exploring the neighbourhood of a potentially efficient solution. On the contrary, an EA usually explores the decision space in a more random way. Thus, a landscape analysis could be interesting to study the bi-objective RSP in more depth.

5 Conclusion

In this paper, a multi-objective routing problem, the bi-objective ring star problem, has been investigated. It has already been studied in a single-objective form where either both objectives have been combined [12] or one objective has been treated as a constraint [16]. Here, for the first time, this problem is formulated

Table 2. Comparison of different metaheuristics for the I_H^- and the R2 metrics by using a Mann-Whitney statistical test with a p-value of 5%. According to the metric under consideration, either the results of the algorithm located at a specific row are significantly better than those of the algorithm located at a specific column ($\succ$), either they are worse ($\prec$), or there is no significant difference between both ($\equiv$).

		I_H^-			R2		
		IBMOLS	IBEA	NSGA-II	IBMOLS	IBEA	NSGA-II
eil51	IBMOLS	-	$\succ$	$\succ$	-	$\succ$	$\succ$
	IBEA	$\prec$	-	$\succ$	$\prec$	-	$\succ$
	NSGA-II	$\prec$	$\prec$	-	$\prec$	$\prec$	-
st70	IBMOLS	-	$\equiv$	$\succ$	-	$\succ$	$\succ$
	IBEA	$\equiv$	-	$\succ$	$\prec$	-	$\succ$
	NSGA-II	$\prec$	$\prec$	-	$\prec$	$\prec$	-
kroA100	IBMOLS	-	$\equiv$	$\succ$	-	$\equiv$	$\succ$
	IBEA	$\equiv$	-	$\succ$	$\equiv$	-	$\succ$
	NSGA-II	$\prec$	$\prec$	-	$\prec$	$\prec$	-
bier127	IBMOLS	-	$\succ$	$\succ$	-	$\succ$	$\succ$
	IBEA	$\prec$	-	$\succ$	$\prec$	-	$\succ$
	NSGA-II	$\prec$	$\prec$	-	$\prec$	$\prec$	-
kroA150	IBMOLS	-	$\succ$	$\succ$	-	$\succ$	$\succ$
	IBEA	$\prec$	-	$\succ$	$\prec$	-	$\succ$
	NSGA-II	$\prec$	$\prec$	-	$\prec$	$\prec$	-
kroA200	IBMOLS	-	$\succ$	$\succ$	-	$\succ$	$\succ$
	IBEA	$\prec$	-	$\equiv$	$\prec$	-	$\equiv$
	NSGA-II	$\prec$	$\equiv$	-	$\prec$	$\equiv$	-
pr264	IBMOLS	-	$\equiv$	$\succ$	-	$\prec$	$\prec$
	IBEA	$\equiv$	-	$\succ$	$\succ$	-	$\succ$
	NSGA-II	$\prec$	$\prec$	-	$\succ$	$\prec$	-
pr299	IBMOLS	-	$\equiv$	$\succ$	-	$\prec$	$\prec$
	IBEA	$\equiv$	-	$\succ$	$\succ$	-	$\succ$
	NSGA-II	$\prec$	$\prec$	-	$\succ$	$\prec$	-

in such a way that multiple conflicting criteria have to be optimized simultaneously. Three metaheuristics have been proposed to approximate the minimal complete set of efficient solutions: a population-based local search with variable neighbourhood and two evolutionary algorithms. Experiments were conducted using various test instances. We concluded that the local search method was significantly more efficient than the evolutionary algorithms on a large majority of instances, with respect to the performance metrics we used. The only instances for which the local search was outperformed were large ones. As a next step, we will try to solve ring star problem instances involving an even bigger number of nodes, to verify if our observations are still valid. If it is the case, it could be interesting to design a cooperation scheme between two different methods (*i.e.* the local search procedure and an evolutionary algorithm) in order to benefit from the respective quality of each one of them. The resulting hybrid metaheuristic could be particularly effective for solving large size problems.

References

1. M. Basseur and E. K. Burke. Indicator-based multi-objective local search. In *CEC'2007*, pages 3100–3107, Singapore, 2007.
2. J. Bean. Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*, 6(2):154–160, 1994.
3. S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler. PISA — a platform and programming language independent interface for search algorithms. In *EMO'2003*, volume 2632 of *LNCS*, pages 494–508, Faro, Portugal, 2003. Springer-Verlag.
4. E. K. Burke and G. Kendall, editors. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer, 2005.
5. J. R. Current and D. A. Schilling. The median tour and maximal covering tour problems: Formulations and heuristics. *European Journal of Operational Research*, 73:114–126, 1994.
6. K. Deb. Multi-objective optimization. In *[4]*, chapter 10, pages 273–316. 2005.
7. K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
8. K. Doerner, A. Focke, and W. J. Gutjahr. Multicriteria tour planning for mobile healthcare facilities in a developing country. *European Journal of Operational Research*, 179(3):1078–1096, 2007.
9. F. W. Glover and G. A. Kochenberger, editors. *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*. Kluwer Academic Publishers, Boston, USA, 2003.
10. M. P. Hansen and A. Jaszkiewicz. Evaluating the quality of approximations of the non-dominated set. Technical Report IMM-REP-1998-7, Institute of Mathematical Modeling, Technical University of Denmark, 1998.
11. J. Knowles, L. Thiele, and E. Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. Technical report, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, 2006. (revised version).
12. M. Labbé, G. Laporte, I. Rodríguez Martín, and J. J. Salazar González. The ring star problem: Polyhedral analysis and exact algorithm. *Networks*, 43:177–189, 2004.
13. A. Liefooghe, M. Basseur, L. Jourdan, and E.-G. Talbi. ParadisEO-MOEO: A framework for evolutionary multi-objective optimization. In *EMO'2007*, volume 4403 of *LNCS*, pages 386–400, Matsushima, Japan, 2007. Springer-Verlag.
14. K. Miettinen. *Nonlinear Multiobjective Optimization*, volume 12. Kluwer Academic Publishers, Boston, USA, 1999.
15. G. Reinelt. TSPLIB – A traveling salesman problem library. *ORSA Journal on Computing*, 3(4):376–384, 1991.
16. J. Renaud, F. F. Boctor, and G. Laporte. Efficient heuristics for median cycle problems. *Journal of the Operational Research Society*, 55(2):179–186, 2004.
17. E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *PPSN'2004*, volume 3242 of *LNCS*, pages 832–842, Birmingham, UK, 2004. Springer-Verlag.
18. E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.
19. E. Zitzler, L. Thiele, M. Laumanns, C. M. Foneseca, and V. Grunert da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.